



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 157 за 2018 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Бондарев А.Е., Кувшинников А.Е.

Применение технологии
DVM/DVMH для решения
параметрических задач
вычислительной газовой
динамики

Рекомендуемая форма библиографической ссылки: Бондарев А.Е., Кувшинников А.Е. Применение технологии DVM/DVMH для решения параметрических задач вычислительной газовой динамики // Препринты ИПМ им. М.В.Келдыша. 2018. № 157. 16 с. doi:[10.20948/prepr-2018-157](https://doi.org/10.20948/prepr-2018-157)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-157>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

А.Е. Бондарев, А.Е. Кувшинников

**Применение технологии DVM/DVMH
для решения параметрических задач
вычислительной газовой динамики**

Москва — 2018

Бондарев А.Е., Кувшинников А.Е.

Применение технологии DVM/DVMH для решения параметрических задач вычислительной газовой динамики

Работа представляет исследование эффективности применения технологии DVM/DVMH для организации параллельного решения параметрических задач газовой динамики. Приведено описание проведенных численных экспериментов по решению тестовых одномерных и двумерных параметрических задач газовой динамики при вариации задаваемого числа вычислительных узлов, параметров этих узлов и выделяемых графических сопроцессоров. Представлены результаты расчетов для ряда тестовых задач в виде характеристик эффективности и ускорения.

Ключевые слова: параметрические задачи, газовая динамика, параллельные вычисления, DVM/DVMH

Alexander Evgenyevich Bondarev, Artem Evgenyevich Kuvshinnikov

Application of DVM / DVMH technology for solving parametric problems of computational gas dynamics

The work presents a study of the effectiveness of DVM / DVMH technology for organizing a parallel solution of parametric problems of gas dynamics. A description is given of the numerical experiments carried out to solve test one-dimensional and two-dimensional parametric problems of gas dynamics with the variation of the given number of computational nodes, the parameters of these nodes, and the allocated graphics coprocessors. The results of calculations for a number of test problems in the form of performance and acceleration characteristics are presented.

Key words: parametric problems, gas dynamics, parallel computing, DVM/DVMH

Авторы выражают глубокую благодарность В.А. Бахтину за плодотворные консультации, советы и замечания.

Введение

В настоящее время параллельные вычисления крайне активно развиваются. Идет соревнование стран, какая страна первой сможет создать эксафлопсный суперкомпьютер (10^{18} операций над числами с плавающей точкой в секунду). Для создания подобного суперкомпьютера необходимы новые энергоэффективные технологии [1], в число которых входит использование графических ускорителей и сопроцессоров. К примеру, в списке TOP500 [2], рейтинге самых мощных вычислительных систем, в первой десятке 8 суперкомпьютеров используют ускорители. А в рейтинге GREEN500 [3], наиболее энергоэффективных суперкомпьютеров, наиболее энергоэффективный суперкомпьютер без ускорителей, Sunway TaihuLight, занимает только 24 место. Те же тенденции прослеживаются и в российском сегменте. В российском аналоге списка TOP500 под названием TOP50 [4] среди самых производительных суперкомпьютеров России и стран СНГ 18 систем используют графические процессоры и 9 систем — ускорители Intel Xeon Phi.

Постоянно появляющиеся новые технологии требуют изменения программного кода для более эффективного использования вычислительной мощности. Долговременная поддержка программ с внедрением новых технологий крайне затратная, поэтому появляются способы автоматического (или полуавтоматического) распараллеливания.

Разработанная в Институте прикладной математики им. М.В. Келдыша РАН высокоуровневая модель параллельного программирования DVMH (DVM for Heterogeneous systems) [5] предназначена для разработки параллельных программ численного моделирования на языках C-DVMH и Fortran-DVMH. Эти языки используют единую модель параллельного программирования (DVMH-модель) и являются расширением стандартных языков Си и Фортран спецификациями параллелизма, оформленными в виде директив компилятору. Поскольку директивы невидимы для стандартных компиляторов, программист может иметь одну программу и для последовательного и для параллельного выполнения на ЭВМ разной архитектуры. Таким образом, единожды вставив в код последовательной программы DVMH-директивы (специальные комментарии), мы получаем параллельную программу.

DVMH-модель [6–8] позволяет создавать эффективные параллельные программы (DVMH-программы) для гетерогенных вычислительных кластеров, в узлах которых в качестве вычислительных устройств наряду с универсальными многоядерными процессорами могут использоваться ускорители (графические процессоры или сопроцессоры Intel Xeon Phi). При этом отображенные на узел вычисления могут автоматически распределяться между вычислительными устройствами узла с учетом их производительности. Компиляторы языков C-DVMH и Fortran-DVMH преобразуют входную программу в параллельную программу, использующую стандартные технологии программирования MPI, OpenMP и CUDA. В состав DVM-системы

входят средства функциональной отладки и отладки эффективности DVMH-программ. Наличие такого эффективного инструмента для распараллеливания программ позволяет решать широчайший спектр задач.

Одним из важнейших и по сей день не оцененных в достаточной степени преимуществ параллельных вычислений является то, что они позволяют параллельно решать в многозадачном режиме одну и ту же задачу с разными входными данными. Подобное преимущество делает параллельные вычисления важнейшим инструментом для решения оптимизационных задач, обратных задач в оптимизационной постановке и задач параметрического поиска. Эти задачи выходят на первый план в современных исследованиях. Характерной общей чертой этих типов задач является то, что в вычислительном смысле они сводятся к массовому решению однотипных задач с изменяющимися входными параметрами. Это обстоятельство делает их удобными объектами для параллельных вычислений на основе многозадачного параллелизма. При решении подобных задач результаты представляют собой большие многомерные объемы данных, анализ и обработка которых, в свою очередь, также проводится с помощью параллельных вычислений. Без применения параллельных вычислений решение задач подобного типа не представляется возможным. Особенности задач подобного типа и применение параллельных вычислений к данным типам задач подробно описываются в работах [9-13].

Продолжая исследования работ [9-13], данная работа имеет своей целью описание результатов проведенных вычислительных экспериментов параллельного решения параметрических задач газовой динамики с использованием технологии DVMH на гибридном вычислительном кластере.

Для исследований эффективности применения системы DVM/DVMH был выбран ряд тестовых задач. Все эти задачи рассматривались как параметрические, то есть для определяющих (ключевых) параметров задачи в заданных диапазонах задавалось разбиение, и затем задачи решались в параллельном многозадачном режиме с помощью системы DVMH. Все выбранные задачи являются простыми и достаточно известными одномерными или двумерными задачами вычислительной газовой динамики. Рассмотрим эти задачи вместе с алгоритмами решения и полученные результаты.

1. Линейное уравнение Бюргерса

Рассмотрим краевую задачу, основанную на применении линейного уравнения Бюргерса, представленного в следующем виде:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2} + f(x, t)$$

Начальные и граничные условия представляются в виде:

$$u(x, t_0) = y(x, t_0),$$

$$u(0,t) = y(0,t), \quad u(L,t) = y(L,t),$$

где $y(x,t)$ — точное решение, описывающее функцию типа бегущей волны.

Точное решение $y(x,t)$ имеет вид:

$$y(x,t) = y(\xi) = e^{-d(1-2\xi)^2}, \quad \text{где } \xi = \frac{t-x}{t_1}.$$

Величины t_1 и d характеризуют ширину и крутизну волны и являются параметрами задачи.

Функция $f(x,t)$ в правой части является выражением вида:

$$f(x,t) = -\frac{\varepsilon}{t_1^2} [16d^2(1-2\xi)^2 - 8d] e^{-d(1-2\xi)^2}.$$

Для решения данной задачи использовалась неявная конечно-разностная схема, подробно описанная в [14,15]. При решении параметрической задачи параметры ε (коэффициент вязкости) и весовой коэффициент гибридной разностной схемы s_k [15] разбивались в определенных диапазонах, и для каждой пары значений решалась описанная выше задача. При проведении экспериментов по распараллеливанию данного программного кода с помощью DVM варьировались следующие параметры:

N — число MPI процессов,

PPN — число MPI процессов на один вычислительный узел.

В проведенных экспериментах число MPI процессов N варьировалось от 1 до 32, а число MPI процессов на один вычислительный узел PPN варьировалось от 1 до 8. Результаты представлены ниже в таблице 1.

Здесь и далее в таблицах будем использовать следующие обозначения:

N — число MPI-процессов,

PPN — число MPI-процессов, запускаемых на одном узле,

THR — число нитей, используемых каждым MPI-процессом,

$CUDA$ — число графических ускорителей, используемых каждым MPI-процессом,

T — время в секундах, S — ускорение T/T_{serial} (или T/T_1),

E — эффективность параллелизации, определяемая как $S/(N*THR)$ (S/N при $THR=0$).

Таблица 1

Уравнение Бюргера — неявная схема

N	PPN	T	S	E
serial	-----	51,4	1	1
1	1	51,4	1,001	1,001
2	1	25,7	1,997	0,998
2	2	25,7	1,998	0,999
4	1	12,9	3,988	0,997

4	2	12,9	3,987	0,997
4	4	12,9	3,991	0,998
8	1	7,61	6,753	0,844
8	2	6,71	7,662	0,958
8	4	6,73	7,641	0,955
8	8	6,98	7,363	0,920
16	2	4,09	12,580	0,786
16	4	3,64	14,129	0,883
16	8	3,76	13,652	0,853
32	2	2,08	24,684	0,769
32	4	2,09	24,613	0,769
32	8	2,16	23,782	0,743

Полученные результаты расчетов показали эффективность реализованного решения по многозадачному распараллеливанию. Одновременно в работе [14] отмечалось, что применяемая в программном коде решения уравнения Бюргера неявная конечно-разностная схема (WW-схема) неудобна для распараллеливания с помощью подключения графических сопроцессоров. Для проведения следующей серии численных тестов ее было необходимо заменить на явный аналог с сохранением свойств аппроксимации и устойчивости. Для проведения дальнейших экспериментов по подключению графических сопроцессоров была проведена замена используемой в алгоритме конечно-разностной схемы. Неявная конечно-разностная схема [15] (WW-схема) была заменена в алгоритме и программном коде на явную схему типа Лакса-Вендроффа, имеющую второй порядок по пространству и времени:

$$u_i^{n+1} = u_i^n - \frac{\tau}{2h} (u_{i+1}^n - u_{i-1}^n) + \frac{\tau^2}{2h^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \varepsilon \frac{\tau}{h^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \tau f(x, t).$$

Для подобной конечно-разностной схемы был реализован программный код с применением директив как DVM, так и DVMH с подключением графических сопроцессоров.

Проведены исследования с увеличением числа узлов разбиения по пространству. Исследовалось влияние на эффективность и ускорение изменения параметров N, PPN, THR, CUDA. В таблице 2 приведены результаты численных экспериментов при следующих значениях параметров: узлов по пространству — 1501, шагов по времени — 7000, шагов по вязкости — 10.

Таблица 2

Уравнение Бюргера — явная схема, 1501 узел

N	PPN	THR	CUDA	T	S	E
1	8	0	0	8,813	1	1
2	8	0	0	8,189	1,0760	0,5381

N	PPN	THR	CUDA	T	S	E
4	8	0	0	7,588	1,1614	0,2904
8	8	0	0	7,182	1,2271	0,1534
16	8	0	0	7,35	1,1991	0,0749
1	1	0	1	12,372	0,7123	
1	1	2	0	7,467	1,1803	0,5901
1	2	4	0	6,459	1,3645	0,2951
2	2	4	0	6,063	1,4536	0,1475

Аналогичные результаты приведены в таблице 3 при следующих значениях параметров: узлов по пространству — 15001, шагов по времени— 7000, шагов по вязкости— 10.

Таблица 3

Уравнение Бюргера — явная схема, 15001 узел

N	PPN	THR	CUDA	T	S	E
1	8	0	0	40,539	1	1
2	8	0	0	24,327	1,6664	0,8332
4	8	0	0	15,915	2,5472	0,6368
8	8	0	0	11,255	3,6019	0,4502
16	8	0	0	9,347	4,3371	0,2711
32	8	0	0	8,324	4,8701	0,1522
1	1	2	0	23,549	1,7215	0,8607
1	2	4	0	14,45	2,8055	0,7014
2	2	4	0	10,058	4,0305	0,5038
4	2	4	0	9,478	4,2772	0,2673
8	2	4	0	8,41	4,8203	0,1506
1	1	0	1	12,799	3,1674	
2	1	0	1	18,251	2,2212	

Продолжая увеличивать число узлов сеточного разбиения по пространственной переменной, получаем результаты, представленные в таблице 4 для параметров: узлов по пространству — 150001, шагов по времени — 7000, шагов по вязкости — 10.

Таблица 4

Уравнение Бюргера — явная схема, 150001 узел

N	PPN	THR	CUDA	T	S	E
1	8	0	0	352,89	1	1
2	8	0	0	184,334	1,9144	0,9572
4	8	0	0	96,38	3,6614	0,9154
8	8	0	0	51,233	6,8879	0,8610

N	PPN	THR	CUDA	T	S	E
16	8	0	0	29,35	12,0235	0,7515
32	8	0	0	18,354	19,2269	0,6008
64	8	0	0	12,799	27,5717	0,4308
128	8	0	0	10,119	34,8740	0,2725
1	2	4	0	94,756	3,7242	0,9311
2	2	4	0	50,522	6,9849	0,8731
4	2	4	0	29,53	11,9502	0,7469
8	2	4	0	18,418	19,1601	0,5988
16	2	4	0	12,959	27,2313	0,4255
32	2	4	0	10,503	33,5990	0,2624
1	1	0	1	19,53	18,0691	
2	1	0	1	21,226	16,6254	

По результатам можно сделать вывод о том, что на грубой сетке использование графических сопроцессоров не только не дает никакого эффекта, но даже замедляет выполнение тестового расчета. При резком увеличении количества узлов расчетной сетки обеспечивается эффективная работа параллельных узлов, при этом подключение графических сопроцессоров также позволяет резко повысить скорость расчетов.

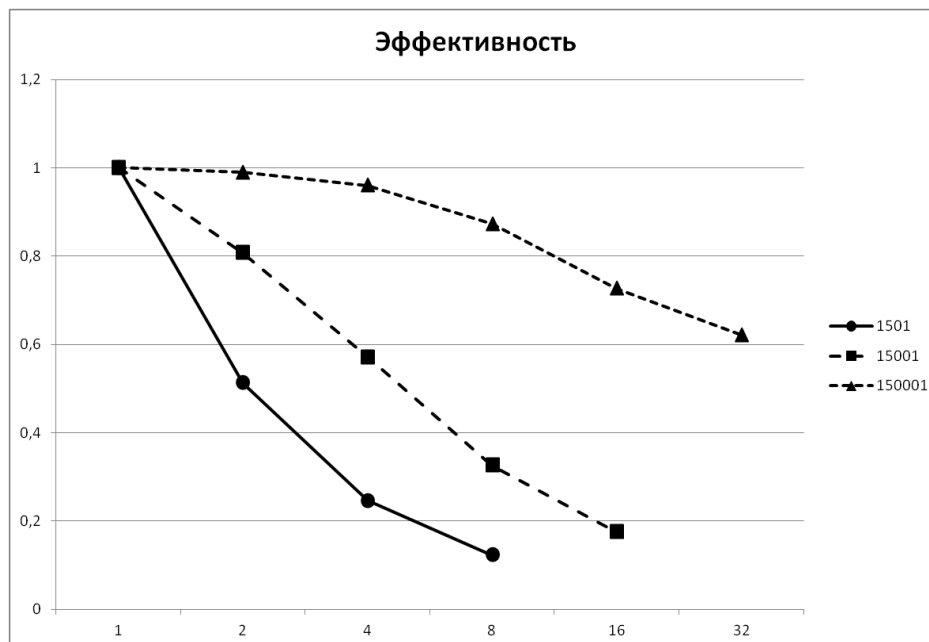


Рис. 1. Значение эффективности при увеличении числа узлов для различного количества узлов расчетной сетки.

На рисунках 1 и 2 представлены сводные результаты расчетных тестов для решения линейного уравнения Бюргера. На рисунке 1 представлены графики зависимости эффективности от увеличения числа узлов вычислительного кластера для различного количества узлов расчетной сетки. На рисунке 2

представлены графики зависимости ускорения от увеличения числа узлов вычислительного кластера для различного количества узлов расчетной сетки. Расчеты ускорения представлены для трех вариантов, четвертый вариант (верхний) соответствует «идеальному» ускорению, когда ускорение происходит во столько раз, сколько процессоров выделено.

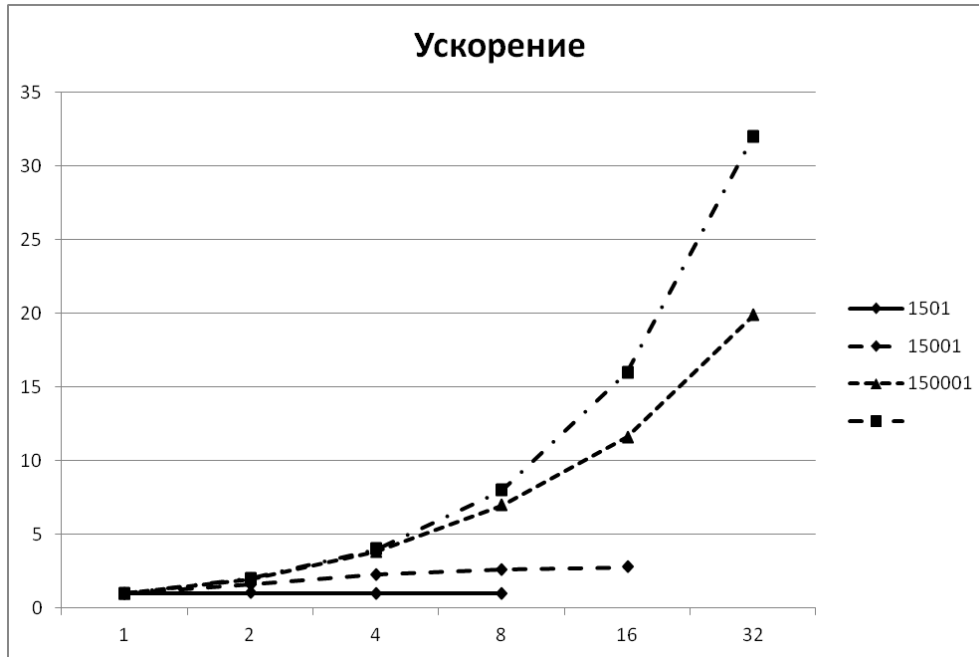


Рис. 2. Значение ускорения при увеличении числа узлов для различного количества узлов расчетной сетки.

2. Квазилинейное уравнение Бюргера

По аналогичной схеме были проведены тестовые расчеты для квазилинейного уравнения Бюргера:

$$\frac{\partial u}{\partial t} + \frac{\partial(u^2/2)}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}.$$

Решением здесь является сглаженная ударная волна:

$$u = \frac{a + b \exp\left(\frac{a-b}{2\varepsilon}(x - x_0 - Dt)\right)}{1 + \exp\left(\frac{a-b}{2\varepsilon}(x - x_0 - Dt)\right)},$$

где $D = (a + b)/2$.

Задача решалась при следующем выборе параметров: $a = 1,0$, $b = 0,2$, $\varepsilon = 0,05$, $x_0 = 5,0$.

В качестве сеточного разбиения задавалось: узлов по пространству — 801, шагов по времени — 9000. Аналогично предыдущей задаче проводилось

увеличение сеточного разбиения по пространству до 1000001 узлов. Результаты представлены на таблицах 5 и 6.

Таблица 5

Квазилинейное уравнение Бюргера — явная схема, 801 узел

N	THR	CUDA	T	S	E
serial			0,5354	1	1
1	0	0	0,6344	0,844	0,844
2	0	0	1,0371	0,5163	0,2581
4	0	0	2,3362	0,2292	0,0573
1	3	0	1,2987	0,4123	0,1374
1	6	0	1,2801	0,4183	0,0697
1	0	1	6,9457	0,0771	
1	0	2	7,2463	0,0739	

Таблица 6

Квазилинейное уравнение Бюргера — явная схема, 100001 узел

N	PPN	THR	CUDA	T	S	E
1	1	0	0	36,7523	1	1
2	1	0	0	16,7374	2,1958	1,0979
2	2	0	0	17,2453	2,1312	1,0656
4	2	0	0	9,2772	3,9616	0,9904
4	4	0	0	9,8062	3,7479	0,937
8	4	0	0	5,8577	6,2742	0,7843
8	8	0	0	8,2898	4,4334	0,554
1	1	2	0	18,1995	2,0194	1,0097
1	1	4	0	10,6017	3,4666	0,8667
1	1	6	0	8,8302	4,1621	0,6937
1	1	8	0	8,1091	4,5322	0,5665
1	1	0	1	9,8264	3,7402	
1	1	0	2	9,6775	3,7977	

Из полученных результатов можно сделать общий вывод. Использование подключения графических сопроцессоров дает резкий эффект ускорения. Но это происходит только в случае достаточно большого количества расчетных узлов. Для грубой сетки, то есть для малого количества расчетных узлов сетки, распараллеливание на несколько процессоров, а тем более подключение графических сопроцессоров приводит к замедлению расчетов по сравнению с последовательным вариантом. Таким образом, расчетная сетка должна быть настолько подробной, чтобы обеспечивать загрузку графических сопроцессоров.

3. Одномерные уравнения газовой динамики

Решалась начально-краевая задача для уравнений динамики невязкого газа (уравнения Эйлера). Исходные уравнения для одномерного течения записывались в консервативной форме:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial x} = 0,$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial (\rho v^2 + p)}{\partial x} = 0,$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial ((\rho E + p)u)}{\partial x} = 0.$$

Здесь $\rho E = \rho u^2 / 2 + \rho \varepsilon$ – плотность полной энергии газа, ε – внутренняя энергия газа, $p = \rho \varepsilon (\gamma - 1)$ – давление газа, γ – показатель адиабаты газа.

Вектор начальных условий:

$$(\rho, u, p)^T = \begin{cases} (1.0, 0.75, 0.1)^T, & x \leq 2.0 \\ (0.125, 0.0, 0.1)^T, & x > 2.0 \end{cases}$$

В подобной постановке данная задача является задачей о распаде произвольного разрыва (задачей Сода) [16,17], для которой есть точное решение.

Рассматривается отрезок [0,8], время расчета $t = 2.4$, шаг по пространству $h = 0.001$, шаг по времени $\tau = 0.0002$.

Для решения данной системы в используемом программном коде применяется схема типа WENO (взвешенные, существенно не осциллирующие схемы), имеющая 5 порядок аппроксимации по пространственной переменной и 3-й порядок аппроксимации по времени [18,19].

Результаты численных экспериментов представлены в таблице 7.

Таблица 7

Задача о распаде произвольного разрыва

N	PPN	THR	CUDA	T	S	E
serial				246,14	1	1
1	12	0	0	259,46	0,95	0,95
2	12	0	0	133,65	1,84	0,92
4	12	0	0	69,79	3,53	0,88
8	12	0	0	38,82	6,34	0,79
12	12	0	0	27,58	8,93	0,74
24	12	0	0	17,6	13,99	0,58
1	1	4	0	69,34	3,55	0,89
1	1	8	0	38,18	6,45	0,81

N	PPN	THR	CUDA	T	S	E
2	2	6	0	27,58	8,93	0,74
1	1	0	1	25,75	9,56	
2	2	0	1	29,84	8,25	

На рисунке 3 представлена величина ускорения для данной задачи при вариации числа процессов и подключении CUDA.



Рис. 3. Зависимость ускорения для задачи Сода от числа процессов подключения CUDA.

4. Уравнения Навье–Стокса (2D)

Рассматривалась задача вязкого течения в канале с заданием следующих граничных условий для решения полной системы уравнений Навье-Стокса для сжимаемого течения. На верхней границе задавалось условие жесткой стенки, на нижней границе — условие оси симметрии, на задней границе — условие экстраполяции, на входной границе — профиль скорости ламинарного пограничного слоя. В качестве начальных данных брались условия входной границы. Решалась система уравнений Навье-Стокса, варьировались три определяющих параметра задачи:

- 1) число Маха (1.5, 2.0, 2.5, 3.0);
- 2) число Рейнольдса (1000, 3162.278, 10000);
- 3) число Прандтля (0.72, 0.8, 0.9, 1.0).

Так как для решения данной параметрической задачи использовалась неявная схема [15], при решении рассматривалось только увеличение числа

процессов аналогично линейному уравнению Бюргера, описанному в разделе 1. Результаты представлены в таблице 7.

Таблица 7

Уравнения Навье–Стокса

N	PPN	THR	CUDA	T	S	E
1	12	0	0	6299,9	1	1
2	12	0	0	3171,29	1,987	0,993
4	12	0	0	1645,51	3,829	0,957
8	12	0	0	868,25	7,256	0,907
12	12	0	0	586,96	10,733	0,894
24	12	0	0	294,45	21,396	0,892
48	12	0	0	168,48	37,392	0,779

На рисунке 4 представлена зависимость ускорения для данной задачи от числа процессов N.

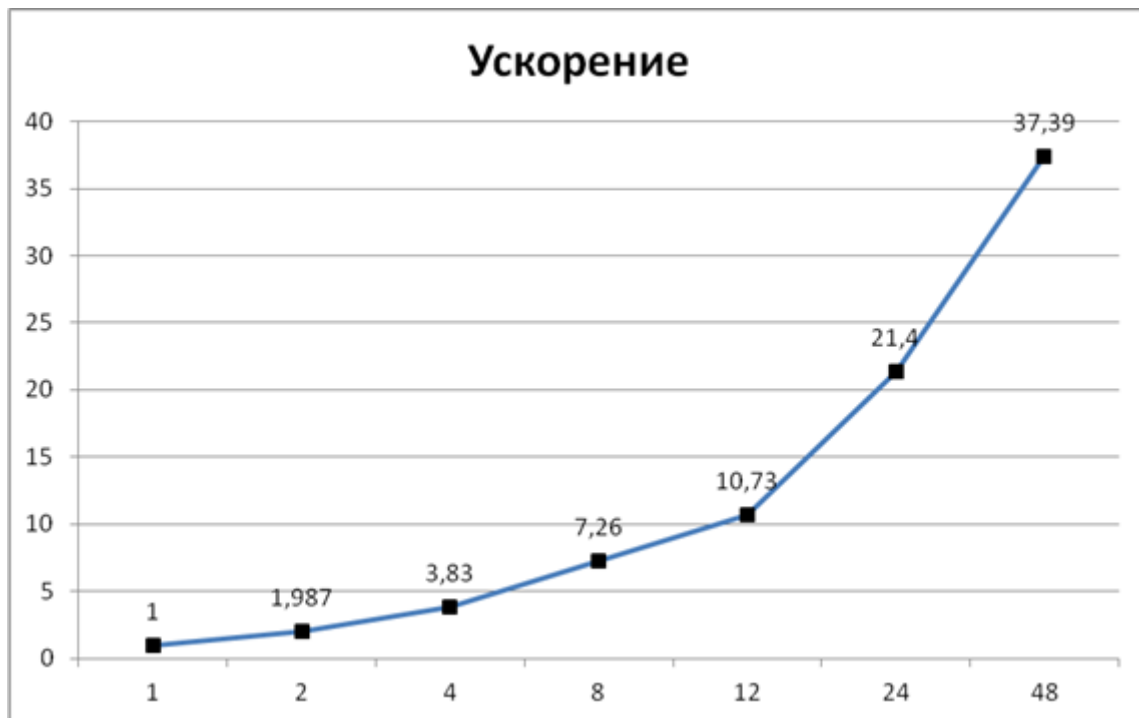


Рис. 4. Зависимость ускорения для уравнений Навье–Стокса от числа процессов.

Заключение

По итогам проведенных экспериментов следует заметить, что система DVM является эффективным инструментом, позволяющим полноценно использовать функционал организации параллельных вычислений с минимальными трудозатратами со стороны пользователя. Стоит заметить, что система вынуждена переводить фортрановские коды на C++, чтобы избежать использования платных версий CUDA-Fortran. Это накладывает ограничения на структуру программы, такие как обязательное использование `pure` функций и невозможность использования массивов без явного обозначения индексов. Если отбросить все эти проблемы и учесть, что приведенные выше задачи являются не настолько вычислительно сложными, как реальные 3D задачи, можно сказать, что DVM-система показывает высокую эффективность для решения параметрических задач газовой динамики.

По результатам проведенных расчетов можно сделать следующий вывод. При подключении графических ускорителей с помощью технологии DVMH возрастают накладные расходы на передачу информации. Поэтому выигрыш можно получить только в задачах с большой вычислительной сложностью. Для получения более точной информации предполагается проведение аналогичных численных экспериментов для более сложных 2D и 3D задач.

Библиографический список

1. Reed D.A., Dongarra J. Exascale Computing and Big Data, *Communications of the ACM*. 2015. Vol. 58. Issue 7. P. 56–68. doi:10.1145/2699414
2. TOP500. URL: <https://www.top500.org> (Дата обращения – 15.07.18)
3. GREEN500. URL: <https://www.top500.org/green500/> (Дата обращения – 15.07.18)
4. Суперкомпьютеры Top50. URL: <http://top50.supercomputers.ru> (Дата обращения – 15.07.18)
5. DVM-система. URL: <http://dvm-system.org>. (Дата обращения – 15.07.18)
6. Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Сазанов Ю.Л. Расширение DVM-модели параллельного программирования для кластеров с гетерогенными узлами. // Супервычисления и математическое моделирование. Труды XIII Международного семинара / Под ред. Р.М. Шагалиева. – Саров: ФГУП “РФЯЦ-ВНИИЭФ”, 2012, с. 84-91.
7. Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н. Смирнов А.А. Использование языка Fortran DVMH для решения задач гидродинамики на высокопроизводительных гибридных вычислительных системах. *Вестник Южно-Уральского государственного университета*,

серия "Вычислительная математика и информатика", том №2, выпуск №3 – Челябинск: Издательский центр ЮУрГУ, 2013, с 106-120.

8. Бахтин В.А., Жукова О.Ф., Катаев Н.А., Колганов А.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Савицкая О.А., Смирнов А.А. Автоматизация распараллеливания программных комплексов // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19-24 сентября 2016 г., г. Новороссийск), М.: ИПМ им. М.В.Келдыша, 2016. С. 76-85.
9. Bondarev A.E, Galaktionov V.A. Parametric Optimizing Analysis of Unsteady Structures and Visualization of Multidimensional Data, *International Journal of Modeling, Simulation and Scientific Computing*, 2013, V.04, supp01, DOI 10.1142/S1793962313410043.
10. Бондарев А.Е. Локализация пространственно-временных структур в пространстве определяющих параметров с применением параллельных вычислений / *Известия ЮФУ. Технические науки*. –№6, 2012. С.100-105.
11. Разработка и тестирование параллельного алгоритма решения задач оптимизационного анализа / Бондарев А.Е. [и др.] // *Информационные технологии в проектировании и производстве*, 2014, № 3, с. 3-9.
12. Bondarev A.E., Galaktionov V.A. Analysis of Space-Time Structures Appearance for Non-Stationary CFD Problems, *Procedia Computer Science*, 51, 1801–1810 (2015).
13. Bondarev A.E. Design and analysis of multidimensional parametrical solutions for time-dependent CFD problems, *Mathematica Montisnigri*, XXXIII, 58-68 (2015).
14. Разработка инструментального программного средства Burgers2 для оптимизации гибридных разностных схем / Бондарев А.Е. [и др.] // Препринты ИПМ им.М.В.Келдыша. 2012. № 53. 12 с.
15. Bondarev A.E. On hybrid numerical method for 2d viscous flows, *Mathematica Montisnigri*, Vol. XXIX, 2014, p. 59-67.
16. Sod G.A. Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws, *J. Comput. Phys.*, 27(1978), 1–31.
17. Toro E. F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag, 1999.
18. Liu X.-D., Osher S., Chan T. Weighted essentially non-oscillatory schemes, *Journal of Computational Physics*, 115(1994) 200-212.
19. Shu C.-W. High order weighted essentially non-oscillatory schemes for convection dominated problems, *SIAM Review*, 51(2009) 82-126.

Оглавление

Введение	3
1. Линейное уравнение Бюргерса	4
2. Квазилинейное уравнение Бюргерса	9
3. Одномерные уравнения газовой динамики	11
4. Уравнения Навье–Стокса (2D).....	12
Заключение.....	14
Библиографический список.....	14